

Selenium framework :-

Anikush.....!

Q] What is selenium framework ??

Framework means set of Rules or Guidelines, that every Automation test Engineer should follow while automating the application.

OR

Framework is the collection code structure which makes code maintenance easy and efficient.

Without Framework user may place the "test data" and "Code" at the same location, which is neither reusable nor readable.

Framework produces beneficial outcomes for the users like

- ① **Code Reusability** :- Once written code we can use multiple times.
- ② **consistency** :- Each and Every Automation Engineer will follow same format.
- ③ Reduces coding efforts and reduces the time
- ④ Increases the coding ~~efficient~~ efficiency.

* * Stages of Framework :-

① **Design** :- It is the 1st stage of Framework which is done by Experienced Senior Authority.

② **Implementation** :- Implementation phase done by senior Test Engineers.

③ **Execution** :- Any Automation Test Engineer is responsible for Execution phase.

* Types of Framework :-

There are mainly 3-types of frameworks ie

- 1) Data Driven Framework (DDF)
- 2) Keyword Driven Framework (KDF)
- 3) Hybrid Framework (combination of DDF+KDF)

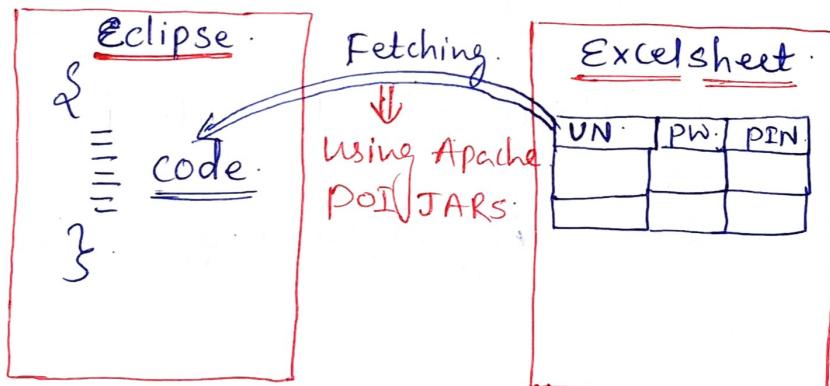
We are implementing DDF framework in our project.

DDF :-

The process of storing data in external files like Excel sheets, CSV files and property files etc. and fetching data using reusable methods is called as "DDF".

DDF will separate test data and test case from each-other. once data is separated from test case, it can be easily modified for further functionality without changing the code.

Ex:-



Ex:1 :- without DDF Example

```
class_without_DDF{  
    Main(){  
        System.setProperty("key", "value");  
        WebDriver d = new ChromeDriver();  
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
        driver.manage().window().maximize();  
        driver.get("http://kite.zerodha.com");  
        // Enter username user name.  
        driver.findElement(By.xpath("//input[@id='userid']")).sendKeys("Dp6458");  
        // Pwd.  
        driver.findElement(By.xpath("//input[@id='password']")).sendKeys("Amol@1234");  
        // Click on login btn.  
        driver.findElement(By.xpath("//button[@class='button-orange-wide']")).click();  
        // Enter pin.  
        driver.findElement(By.xpath("//input[@id='pin']")).sendKeys("171992");  
        // Click on Cont btn.  
        driver.findElement(By.xpath("//button[text()='Continue']")).click();  
        // Verify user id  
        String expuserid = "Dp6458";  
        String actuserid = driver.findElement(By.xpath("//span[@class='userid']")).  
        getAttribute("value");  
        if (actuserid.equals(expuserid)) {  
            System.out.println("Pass");  
        } else {  
            System.out.println("Fail");  
        }  
    }  
}
```

Ex2: with_DDF_Example

Class_with_DDF

main() {

System.setProperty("key", "value");
WD d = new CD();

[Path D:\xls.xls]
at Taskbar right Excel close
↑ (otherwise exception)

FileInputStream fis = new FileInputStream("excel path");
Sheet sh = WorkbookFactory.create(file).getSheet("Sheet1");
↓
(FileNotFoundException)

System.setProperty("key", "value");
WD d = new CD();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.manage().window().maximize();
driver.get("zerodha");

// Enter UN

String UN = sh.getRow(0).getCell(0).getStringCellValue();
driver.findElement(By.xpath("//input[@id='userid']")).sendKeys(UN);

// Enter pwd

String pwd = sh.getRow(0).getCell(1).getStringCellValue();
driver.findElement(By.xpath("//input[@id='pwd']")).sendKeys(pwd);

// click on Login Bttn

driver.findElement(By.xpath("//button[@class='button-orange wide']")).click();

// Enter pin

String pin = sh.getRow(0).getCell(2).getStringCellValue();
driver.findElement(By.xpath("//input[@id='pin']")).sendKeys(pin);

// click on Continue Bttn

driver.findElement(By.xpath("//button[text()='Continue']")).click();

```

    // Verify user id.
    String expuserid = sh.getRow(0).getCell(3).getStringCellValue();
    String actuserid = driver.findElement(By.xpath("//span[@class='userid']")).getText();
    if (actuserid.equals(expuserid)) {
        System.out.println("Pass");
    } else {
        System.out.println("Fail");
    }
}

```

Encapsulation Concept:

Hiding data from other classes using private access specifiers is called as Encapsulation.

- 1) Data members / variables should be declared globally with access level private.
- 2) Initialize within constructor with access level public
- 3) Utilize within the method. with access level public.

POM:- Page object model

It is the Java concept, It is also called as design pattern which is used to design the classes. in Test script. We can use POM in Selenium to design each and every page separately.

If in our project 10 pages are there, then we have to design 10 POM classes.

Rules for POM class:

- ① One webpage can be represented as one POM class
- ② Various elements on webpage are represented as variables of POM class.
- ③ Each and every POM class must ends with word "Page".

** POM with PageFactory :-

POM with page factory strictly follows Encapsulation concept where.

- ① Data members/variables can be declared globally with access level private.
- ② Initialize within constructor as access level public
- ③ Utilize within method as access level public

Conclusion:-

- ① Take variables as private \Rightarrow ~~private WebElement UN;~~
- ② Take constructor as public \Rightarrow ~~public classname (web)~~

Public classname (webdriver driver) { ~~driver~~

Actually webdriver driver is coming from
Test class, as we know that using constructor we
can call ~~one~~ another class.]

- ③ Take method as public \Rightarrow

public void enterUN () {
}

Notes:-

- ① No of data members that need to be created under POM class will depend on no. of elements that need to handle on a webpage.

- ② POM class will not contain main() to run all the POM classes we need one user logic class which contains main() i.e. called as "Test class"

- ③ Test class contains main() and this test class is responsible to run all POM classes.

Ex3:- POM with Pagefactory

class KiteLoginPage {

// Step1 :- Declaration

@FindBy(xpath = " //input[@id='userid']") private WebElement UN;

@FindBy(xpath = " //input[@id='password']") private WebElement PWD;

@FindBy(xpath = " //button[@class='button-orange wide']")

private WebElement loginBtn;

↓ [we declared all variables
as private i.e. Encapsulated
in class]

// Step2 :- Initialization

public KiteLoginPage(WebDriver driver) {

PageFactory.initElements(driver, this); Here const^r is used
i.e. parameterized const^r

Since this const^r is public we can easily provide
driver from Test class. as we know that para-
metrized constructors are used to call ^{BLG} classes in
UIC classes. [WebDriver driver]. driver → WebDriver
Interface

// Step3 :- Implementation / usage

public void enterUN() {

UN.sendKeys("DPG1458");

}

↳ method is created for UN. Because we have to
enter Username for UN text field.

** Working of Pagefactory.initElement(driver, this) ; ; ; →

① Pagefactory is the class which contains initElement as static method.

Actually POM is the solution but for which problem ?

⇒ class {
main() {

System.setProperty("key", "value");

WebDriver d = new WebDriver();

driver.get("http://www.google.com");

WebElement UN = driver.findElement(By.id("username"));

driver.navigate().refresh(); → here if we refresh the page
UN.sendKeys("admin"); it will not write "admin"? Why?

⇒ findElement will give us address of the page, whenever we
refresh the page, address of the page can be changes, but
this new address/ changed address/ latest address will not cap-
tured by findElement hence it gives "stale Element ref-
erence exception"

↓
(first/ old/ former)

Q] How to get new address/ latest address ?

⇒ @FindBy ⇒ this concept is coming from POM.

findElement

[It will give address of page
but after refresh the page
it will not give updated address/
new address]

@FindBy

[It will give latest/updated address
after refresh the page]

↓
So that we get stale Element
reference exception

POM is Java design pattern, It is the solution for problem. Whenever we refresh the page address can be change.. this new address we can fetch using POM concept.

@FindBy ⇒ 1st of all it will take current address of webpage

But once page get refreshed, at that time initElement() from pagefactory class supports to @FindBy annotation. go. Page is refreshed bring new address.

Again page is refreshed one more time again initElement() tells to @FindBy hey! annotation go and bring new address.

like this it will give us new/latest/updated address. hence address never becomes old hence we can't get "stale element reference exception".

Advantages of POM: →

- ① Code reusability
- ② we can avoid "stale element reference exception"
- ③ Existing script ~~become~~ we can maintain easily.
- ④ If tomorrow anyone/dev changes the username then there is no need of changing code inside the test class, we can easily change username from POM class only.
- ⑤ All UN, pwd, LoginBtn, PIN, contBtn we are storing at one place hence it is called "element storage repository/object repository".

Note:- Actually selenium does not have this concept to store elements at one place hence we are using POM which is in Java design pattern.

My convenience:-

- ① What is stale element ref^{le} exceⁿ? \Rightarrow The reference/ address of element is old
- ② Why we get exception?
 \Rightarrow When we use `get`'s to get the address of an element, store it in reference variable, but before performing action, if the webpage is refreshed, then on the webpage elements address can be changed, but in ref variable we still have old address, and we are trying to perform action on ~~old~~ old address only hence we get this exception.

③ How to handle it ?

- \Rightarrow By using POM (Page object model)

It is one of the Java design pattern, mainly it is object repository which is used to store all the web elements in one place hence it is also called as page object repository.

④ How does POM handle stale element reference exceⁿ??

- \Rightarrow In POM we use `@FindBy` to get the address of element.

`@FindBy` annotation will always get new address of the element therefore address will never becomes old, hence we can avoid getting this exception.

⑤ How do we declare elements in POM??

- \Rightarrow POM class means any class which contains `@FindBy` is called as POM class.

In this class we declare elements using `@FindBy`.

Syntax \Rightarrow `@FindBy(locator name = "locator value")` Access spec Return type Element name;

Ex \Rightarrow

`@FindBy(id = "username") private WebElement UNTB;`

Q] How do we initialize elements in POM??

⇒ We use initElements() of PageFactory class to initialize all elements at once.

initElements() gives instructions to ~~all the~~ @FindBy annotation to fetch the address and store it inside reference variable.

initElements takes two arguments:-

① ~~Browser class~~

① WebDriver's reference variable ⇒ means Browser class object reference.

ex:- driver.

② Object page ⇒ means in which class we have to store the elements.

a) If elements are in same class/current class we use "this"

Ex:- PageFactory.initElements(driver, this);

b) If elements are in some other class then, create object of that class and pass reference variable as argument.

Ex:- POMclass pc = new POMclass();

PageFactory.initElements(driver, pc)

Q] Is userdefined constructor is compulsory to initialize the elements??

⇒ No, But initializing elements with initElement() is mandatory.

If not initialised then we will get Null pointer exception

Q] How to handle multiple Elements in POM class?

⇒ use the Set/Type as `List<WebElement>`

Q] How do we utilize in POM class??

⇒ By creating public 'getter' and 'setter' methods.

(a) Address returning method

Public WebElement getUNTB() {

 Return UNTB;

}

(b) Action performing method.

Public void setUsername(string vn) {

~~ENTER~~

 UNTB.sendKeys(vn);

}

Uses of POM:-

- ① It is used to store all the elements in one place. Hence it acts as element Repository/Object Repository.
- ② It can be avoid stale element reference exception.
- ③ Code Reusability is achieved, which increase the efficiency of test script development.
- ④ Maintaining Automation Script is easy [Especially when requirements are frequently changing]
- ⑤ Helps in encapsulating the elements
- ⑥ ~~So~~ we can achieve abstraction
- ⑦ We can achieve method driven Testing.

~~POM~~

Ex3] POM with pagefactory ~~and without DDF~~ Ex:-

class Kiteloginpage {

//Step1:- Declaration

@FindBy(xpath = "//input[@id = 'userId']") private WebElement UN;

@FindBy(xpath = "//input[@id = 'pwd']") private WebElement pwd;

@FindBy(xpath = "//button[@class = 'button-orange wide']") private WebElement loginBtn();

//Step2: Initialization

public Kiteloginpage (WebDriver driver) {
PageFactory.initElements (driver, this);
}.

//Step3: Utilization

public void enterUN () {
UN.sendKeys ("DPG458");
}

public void enterpwd () {
pwd.sendKeys ("Amol@1234");
}

public void clickonLoginBtn () {
loginBtn.click();
}

```
class kiteLogin2page{  
    @FindBy(xpath = "//*[@id='pin']") private WebElement pin;  
    @FindBy(xpath = "//*[@type='submit']") private WebElement ctnBtng;
```

```
public kiteLogin2page(WebDriver driver){  
    PageFactory.initElement(driver, this);  
}
```

```
public void enterpin(){  
    pin.sendKeys("171992");  
}
```

```
public void clickonctnBtn(){  
    ctnBtng.click();  
}
```

```
class kiteHomepage{
```

```
@FindBy(xpath = "//*[@class='userid']") private WebElement user_id;
```

```
public kiteHomepage(WebDriver driver){  
    PageFactory.initElement(driver, this);  
}
```

```
public void verifyuserid(){  
    String expuserid = "171992 Dp5458";  
    String actuserid = user_id.getText();
```

```
if (actuserid.equals(expuserid)) {  
    SysIn("pass");  
}  
else {  
    SysO("Fail");  
}  
}  
}  
}  
}
```

```
class kiteLoginTest {
```

```
    main() {
```

```
        System.setProperty("key", "value");
```

```
        WD d = new CD();
```

```
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

```
        driver.manage().window().maximize();
```

```
        driver.get("https://kite.zerodha.com/");
```

```
        KiteLogin1page login1 = new KiteLogin1page(driver);  
        login1.enterUN();  
        login1.enterPWD();  
        login1.clickonloginBtn();
```

```
        KiteLogin2page login2 = new KiteLogin2page(driver);  
        login2.enterPin();  
        login2.clickonctrnBtn();
```

```

kiteHomepage home = new kiteHomepage(driver);
home.verifyuserid();
Thread.sleep(2000);
driver.close();
}
}

```

Ex4:- POM with Pagefactory using DAF :-

```

class kiteLoginPage {
    @FindBy(xpath = "//input[@id='userid']") private WebElement UN;
    @FindBy(xpath = "//input[@id='pwd']") private WebElement pwd;
    @FindBy(xpath = "//button[@class='button-orangewide'])")
    private WebElement loginBtn;
}

```

```

public kiteLoginPage (WebDriver driver) {
    Pagefactory.initElements(driver, this);
}

```

```

public void inputkiteLoginPageUN (String username) {
    UN.sendKeys(username);
}

```

```

public void inputkiteLoginPagepwd (String password) {
    pwd.sendKeys(password);
}

```

```

public void clickkiteLoginPageLoginBtn () {
    loginBtn.click();
}

```

④ Class kiteLogin2Page {

```
@FindBy(xpath = "//input[@id='pin']") private WebElement pin;  
@FindBy(xpath = "//button[text()='Continue']") private WebElement ctnBtm;
```

```
Public kiteLogin2Page(WebDriver driver) {
```

```
    PageFactory.initElements(driver, this);  
}
```

```
Public void iplkiteLogin2Page(String pinvalue) {
```

```
    pin.sendKeys(pinvalue);  
}
```

```
Public void clickLogin2Page() {
```

```
    ctnBtm.click();  
}
```

```
}
```

⑤ Class kiteHomepage {

```
@FindBy(xpath = "//span[@class='userid']") private WebElement userid;
```

```
Public kiteHomepage(WebDriver driver) {
```

```
    PageFactory.initElements(driver, this);  
}
```

```
Public void verifykiteHomepageuserid(String expuserid) {
```

```
    String actuserid = userid.getText();  
    if (actuserid.equals(expuserid)) {
```

```
    sys0("pass");
}
else{
    sys0("Fail");
}
}
}
```

```
Class KiteloginTest{
```

```
main(){
```

```
FileInputStream fis = new FIS("excelpath");
```

```
Sheet sh = workbookFactory.create(fis).getsheet("DDF");
```

```
System.setProperty("IC", "V");
```

```
WebDriver d = new CD();
```

```
implicitlyWait(10,TimeUnit.SECONDS);
```

```
maximize();
```

```
driver.get("https://zerodha.com");
```

```
Kitelogin1page login1 = new Kitelogin1page(driver);
```

```
String username = sh.getRow(0).getCell(0).getstryCellValue();
```

```
login1.iplkitelogin1pageusername(username);
```

```
String password = sh.getRow(0).getCell(1).getstryCellValue();
```

```
login1.iplkitelogin1pagepwd(password);
```

```
login1.clickkiteloginpagesbtn();
```

Kite home

KiteLoginPage login2 = new KiteLoginPage(driver);

String pinvalue = sh.getRow(0).getCell(2).getStdStringValue();

login2.iplkitelogin2page(pinvalue);

login2.clickLoginPage(btn);

login2.clickKiteLoginPageBtn();

KiteHomepage home = new KiteHomepage(driver);

String expnsesid = sh.getRow(0).getCell(0).getStdStringValue();

home.VerifyKiteLoginPageUserid(expnsesid);

}

↳ [Exposure ID]

DDF Scan

Not provide exact
Pom class in]

}

TestNG → Test Next Generation. [3rd party tool]

→ TestNG can be unit testing tool/ framework.

→ It supports both languages Java and .Net.

→ It is invented by Cedric Beust. (It is 3rd party tool)

Q] Why TestNG Introduce?

⇒ ① We can't run multiple classes & single project ~~in Selenium~~ in Selenium, But using TestNG we can easily run multiple classes & single project with single click. this called as ~~parallel~~ execution. It is done using Test suite concept.

② Report generation is very imp in Automation Testing as well as manual testing, using Selenium we can generate report but using TestNG we can generate emailable report and index.html Report.

* TestNG will generate default report.

Advantages of TestNG:-

① Normal class:-

① we can get the o/p in the form of emailable HTML

② we can get the o/p in ~~in~~ ^{reports} index.html report

③ Perform Batch Execution.

[Running multiple scripts with single click]

④

Rules for TestNG :-

① Never use default package (always create class under some package)

② Never use main() (use @Test)

③ Never use System.out.println (use Reporter.log)

Reporter.log ("abc") \Rightarrow It will print only in ^{emailable} report.

Reporter.log ("abc", true) \Rightarrow It will print in report and console ^{emailable}

Test class :- Any class which contains @Test is called as Test class.

① Emailable Report :-

Report generation is very imp whenever AT is going on.

Because by looking at result we can easily identify that how many test cases are passed and how many failed and skipped.

By looking at the report we come to know what is the status of project is.

Selenium webdrivers is used for ~~the~~ automating the webapplication but it won't generate any report. So we are using TestNG for report generation.

Steps to generate Emailable report :-

1) create a class \rightarrow Execute Test class \rightarrow Refresh the project

2) we will get testng folder

3) In that folder right click on emailable report.html and select the option open with web browser ~~or~~ double click on it emailable report will open in Multiclick

```
Ex1] class Normal_Testclass{  
    main(){  
        System.setProperty ("key", "value");  
        WebDriver d = new CD();  
        driver.manage().window().maximize();  
        driver.manage().timeouts().implicitlyWait (10, TimeUnit.  
        driver.get ("https://kite.zerodha.com/");  
    }  
}
```

Ex2] class TestNG_Testclass {

@Test

```
public void openApp() {
```

```
System.setproperty("key", "value");
```

WOD d = new ^V eD();

```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

drivers. manage () . windows () . maximizer () .
seconds)

```
driver.get("http://www.w3schools.com/html/html_intro.htm");  
driver.maximizeWindow();
```

```
driver.get ("kite.zesokhei");
```

3

3

Ex3] class EmailableReport {

① Test

public void TC1() {

Reporter.log ("running TC1", true);

}

② Test

public void TC2() {

{

Reporter.log ("running TC2", true);

}

}

Steps to generate Emailable Report ??

① 1st create one class and run the class get the otp
(normally)

② Right click on class \Rightarrow TestNG \Rightarrow Convert to TestNG then.

Location: /Freg/Auto/ testing.xml

Suite name:- smite

Test Name:- Test

 []

इसी मानदण्डित रूप से फॉर्मेट
में टेस्ट सूट नाम इसी रूप से

for ex:- abc.xml

इसी फॉर्मेट.

③ Then refresh the package/project so that under testng folder abc.xml यहाँ create होता है। Double click on it
suite create होता है।

④ click on Run button \Rightarrow TestNG suite \Rightarrow OK (Suite Run कराया गया है।
NOT output नहीं)

⑤ इसी testng पर under Freg Emailable-report.html यहाँ किया गया है।
Right click on it open with Web Browser यहाँ।

TestNG Annotations →

Before Emailable report we have to teach this:-

Sys0.

① It will print only in console

Reporter.log

static
log → method

Reporter → class

① Reporter.log("hi");

↳ It will print only in Email report

Reporter.log("hi", true)

↳ It will print in console as well Emailable Report.

@Test → It is annotation.

acts on
Main({})

acts on
TestCase

Advantages of Testing :-

 we can print output in console as well available report.

Sys0	Reporter-Log.
------	---------------

② Emailable Reports:-

Ex:- class ER {

@Test .

`@Test
public void TC1() { } } ← 1st write this then write
@Test`

```
Reporter.log("running TC1", true);
```

3

② Test

```
public void Tc208
```

2

```
Reporter->log ("Running TC2",true);
```

11

⇒ procedure :-

- ① Right click ⇒ TestNG ⇒ Convert To TestNG
⇒ give the suite name.
- ② Refresh Package/Project so that testoutput folder will generate
- ③ under testoutput folder available.xml report is there, right click on it ⇒ open with WebBrowser
↓
In multicolor report will visible.

③ Annotations :- Annotations are inbuilt codes which accepts locators as input.

① @Before class :- It executes before every class.
(open the browser, enters the URL)

② @Before method :- It executes before every Test method.

③ @Test :- This annotation is used for Execution purpose

④ @After method :- It executes after every Test.

⑤ @After class :- It executes after every class

~~@Before~~
~~@Before~~

Actual Execution :-

@Before class (ex:- openBrowser)

@Before method (ex:- App"login")

@Test (main/actual test case)

@After method (ex:- App"logontest")

@After class (ex:- closeBrowser)

Ex :- public class TestNG {

 @Before class

 public void openBrowser () {

 Reporter.log ("--openBrowser", true);

}

~~public~~ @ Before method

```
public void loginToApp() {  
    Reporter.log("Login to app", true);  
}
```

@ Test

```
public void verifyuserid() {  
    Reporter.log("running verifyuserid TC", true);  
}
```

@ After method

```
public void logOutToApp() {  
    Reporter.log("Logout to APP", true);  
}
```

@ After class

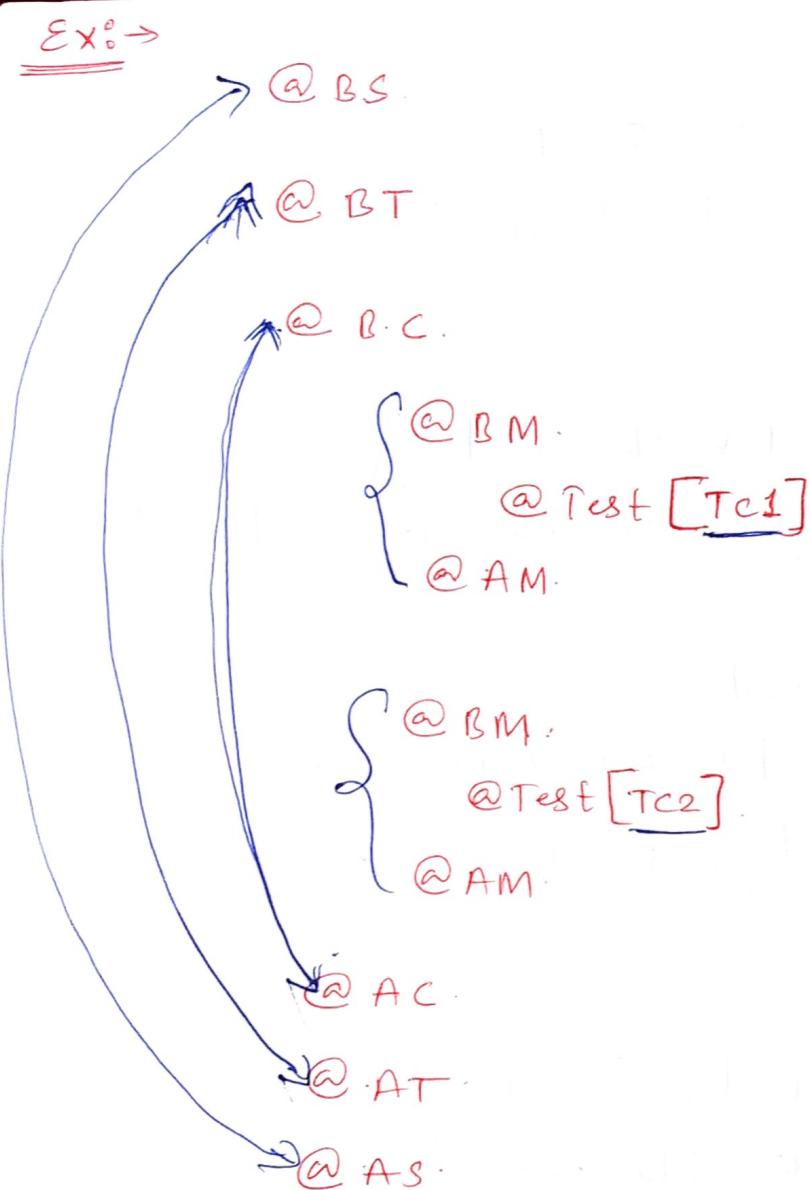
```
public void closeBrowser() {  
    Reporter.log("close browser", true);  
}
```

}

}

Flow of Annotations:

- ① BeforeSuite → execute Before start of suite (Connect to DB)
- ② Before ~~class~~ Test → execute Before every Test Runner
- ③ Before class →
- ④ Before method →
- ⑤ Test →
- ⑥ After method →
- ⑦ After class →
- ⑧ After Test → execute After every Test Runner
- ⑨ After suite → Disconnect from DB



[4] Batch Executor / Test suite :- [by indext 37E]

Right click on the project → TestNG → convert
TO TestNG → Finish

So that .xml file is generated as below which is nothing but Test suite / Batch Executor.

Syntax: →

```

<suite name="suite name">
  <test name="TestName">
    <classes>
      <class name="Testing.sumo"/>
      <class name="Testing.Demo"/>
      <class name="Testing.Remo"/>
    </classes>
  </test>
</suite>
  
```

TestNG Flags / TestNG keywords:-

① These are invented to avoid repetition of code.

① Invocation count:- sometimes same test case we need to execute multiple times. it is possible using invocation count.

Eg:- invocation count = 5; [same TC Runs for 5 times]

Ex:- class A {

 @Test (invocation count = 10)

 public void TC1() {

 Reporter.log("Running TC1", true)

}

② Priority:- we can change the execution order of TC acc. to our comfort.

Ex:- priority = 1. → It will execute 1st.

Priority = 0 → It will execute earlier than 1st.

(+ve, -ve, Duplicate and zero also)

[
 TC with priority same / Duplicate TC with methodname / Testcase with Name
 TC with alphabet order of execute

Ex:- class B {

 @Test (priority = 3)

 public void TC1() {

 Reporter.log("Running TC1", true);

 @Test (priority = 2)

 public void TC2() {

 Reporter.log("Running TC2", true);

 @Test (priority = 1)

 public void TC3() {

 Reporter.log("Running TC3", true);

3 3 //

③ enabled :- If we want to ignore/skip the testcase at that time we use "enabled=false".

Ex:- class C {

 @Test (enabled = false)

④ timeouts :- If one of the testcase is taking too much time for Execution, In such case we can set the time using timeout keyword

Ex:- @Test (timeout = 8000)

⑤ depends on methods :- Sometimes Logout scenario depends on Login scenario, in such case we can follow this keyword.

Ex:- class D {

 @Test

 public void login() {

 Reporter.log("Running login", true);

 }

 @Test (dependsOnMethods = {"login"})

 public void logout() {

 Reporter.log("Running logout", true);

 }

}

→ login → it fail (sim) it depends on methods
different test set.

* sometimes depend on login1 & login2 it depends
3rd test (for ex:- lite-zerodha)

@Test (dependsOnMethods = {"login1", "login2"})

- ⑥ groups:- we can merge group of required test methods
- ⑦ name:- used for identification purpose
- ⑧ parameters:- used to pick specific parameter
- ⑨ dataprovider:- used in Data Driven concept of TestNG
- ⑩ alwaysRun:- If true, then it will run always even if dependant methods are failed.

5]

Batch Execution / Test Suite :-

The process running multiple classes with single click is called as Batch Execution.

For ex:- we have 3 classes ~~having~~ each class having 3 test case.

class Sample1 {	class Sample2 {	class Sample3 {
TC1() {	TC04	TC07
TC2()	TC05	TC08
TC3()	TC06	TC09

Public class Sample1 {

@Test

public void TC1() {

 Reporter.log ("running TC1", true);
}

@Test

public void TC2() {

 Reporter.log ("running TC2", true);
}

@Test

public void TC3() {

 Reporter.log ("running TC3", true);
}

}

Public class Sample2 {

@Test

public void TC4() {

 Reporter.log ("running TC4", true);
}

@Test

public void TC5() {

 Reporter.log ("running TC5", true);
}

② @Test

```
public void TC6(){  
    Reporter.log("running TC6",true);  
}
```

public class Sample3 {

@Test

```
public void TC7(){  
    Reporter.log("running TC7",true);  
}
```

@Test

```
public void TC8(){  
    Reporter.log("running TC8",true);  
}
```

@Test

```
public void TC9(){  
    Reporter.log("running TC9",true);  
}
```

==

Procedure ⇒

- ① ~~control~~ ~~using~~ 3 class select ~~ctrl~~ convert to TestNG.xml
- ② project ~~ctrl~~ refresh ~~ctrl~~, so that test of folder create ~~ctrl~~, ~~ctrl~~ ~~ctrl~~ [Demo.XML] ~~ctrl~~ open ~~ctrl~~ suite open ~~ctrl~~ suite Run ~~ctrl~~
- ③ ~~ctrl~~ ~~ctrl~~ emailable.html ~~ctrl~~ open with web browser ~~ctrl~~ so that pass and fail ~~ctrl~~

Write suite syntax :-

⑥ Disable Test case :-

There are two ways to disable TC:-

- ① Using classes → enabled = false [class level changes]
 - ② Using suite → include/exclude [suite level changes]
- ↓ ↓
 Keyword. Keyword.

Q] How to make changes in suite XML.

⇒ <?xml version = "1.0"?>
<suitename = "suite">
<testthread count = "5" = "Test">
<classes>
 <classname = "TestNG-ex.A"/>
 <classname = "TestNG-ex.B"/>
 <classname = "TestNG-ex.C"/>
</classes>
</test><!--Test-->
</suite><!--suite-->

I want to disable TC3 from A class. i.e

⇒ <classes>
 <classname = "TestNG-ex.A">
 <methods>
 <exclude name = "TC3"/>
 <exclude name = "TC2"/>
 </methods>
 </class>
</classes>

→ ~~TC3~~ / "from A class".

Ex1] Disable TC using Keyword (classlevel changes)

class Sample { // classlevel changes

@Test

```
public void TC1() {  
    Reporter.log("running TC1", true);  
}
```

@Test

```
public void TC2() {  
    Reporter.log("running TC2", true);  
}
```

@Test (enabled=false)

```
public void TC3() {  
    Reporter.log("running TC3", true);  
}
```

Ex2] Disable Test case using Test suite:-

class Sample { // suitelvel changes

@Test

```
public void TC1() {  
    Reporter.log("running TC1", true);  
}
```

@Test

```
public void TC2() {  
    Reporter.log("running TC2", true);  
}
```

@Test

```
public void TC3() {  
    Reporter.log("running TC3", true);  
}
```

====

⇒ suitelevel changes:-

<classes>

<classname = "Testing_ex. ~~Example~~">

↳ don't write "/" here

<methods>

<exclude name = "tc3"/>

</methods>

</class>

</classes>

⑦ Failed.xml file:-

→ It is used to run only failed test cases.

Q] How to execute only failed test case?

⇒ using Failed.xml file.

Q] How to do test case disable?

⇒ using enabled = false.

While executing the Automation scripts, test cases may fail for several reasons. To optimize our next run, we need to re-run only failed test cases.

Steps to execute Failed.xml file:-

① Create testing.xml file under project folder

② Execute testing.xml file.

③ In the test o/p folder testing failed.xml file will be created.

④ Execute "testing^xml" file. In this way we can execute failed tc in Testing.

[Don't do it file or don't create it].

Reasons for fail Test cases →

Q] why test cases gets failed ??

⇒ ① Environmental Issue

② Script Error

③ Bug

Ex:- class Failedxmlfile{

 @ Test

```
    public void TC1() {  
        Reporter.log("running TC1");  
    }
```

 @ Test

```
    public void TC2() {  
        Reporter.log("running TC2");  
    }
```

 @ Test

```
    public void TC3() {  
        Reporter.log("running TC3");  
        Assert.assertFail();  
    }
```

↓ [इस पर TC fail होता है तो Project Refresh करके failed.xml file generate होता है जिसके बारे में failed.xml file में उनकी TC bug free कहा रखा जाता है]

⑧] Parallel Testing :- // Parallel Execution

It is process of running multiple testcases parallelly
rather than one after one.

Here we can open two applications simultaneously
on same browser at same time.

Public class parallel1 {

@Test

PVTC1()

System.setp("k", "v");

WD d = new CD();

maximize();

driver.get("zerodha");

Thread.sleep(2000);

driver.close();

}

}

==

Public class parallel2 {

@Test

PVTC2()

System.setp("k", "v");

WD d = new CD();

maximize();

driver.get("fb");

Thread.sleep(4000);

driver.close();

}

}

==

Public class Parallel3 {

@Test

public void TC3() {

System.setProperty("k", "v");

WebDriver d = new WebDriver();

driver.get("http://google");

}

}

Suite =>

<suite name="suite">

<test threadCount="5" name="test">

<classes>

<class name="Testing.c"/>

</classes>

</test> <!-- Test -->

</suite> <!-- suite -->

↓

After changing

<suite name="suite" parallel="tests" >

③

<tests threadCount="5" name="test1">

②

<classes>

<class name="Testing.A"/>

</classes>

</test>

<test threadCount="5" name="test2">

②

<classes>

<class name="Testing.B"/>

</classes>

</test> <!-- Test -->

</suite> <!-- suite -->

① copy & paste

from <test>

to <tests>

② name change

name=Test1

name=Test2

③ class name change

Testing.A

<class name="Testing.c"/>

<class name="Testing.d"/>

④

<suite name="suite" parallel="tests" >

→ 2 ways

add or
remove

* MultiBrowser Testing / compatibility Testing :-

We can run our script on different browsers simultaneously.

Ex:-



Public class MultiBrowser

@ Parameters("browsername") // annotation

@ Test

Private void (String browsername) {

Webdriver driver = null; // global [Runtime polymorphism]

if (browsername.equals("Chrome")) {

System.setProperty("K", "V");

driver = new ChromeDriver();

}

elseif (browsername.equals("Firefox")) {

System.setProperty("K", "V");

}

driver = new FirefoxDriver();

driver.get("https://google.com"); // globally declared

}

It is run directly or run through TestNG
Execute करते हैं। इसे multibrowsing testing कहते हैं।
Similarly direct program run करते हैं या यह suite
Run करते हैं।

* * Steps/modifications for multibrowser Testing.

① Normally psm timer, not suite run at so that suite this classes opt is affirm

Class A .

@ Test.

Public void TC() {

```
System.setProperty("k", "v");
WD d=new CD();
driver.get("google");
```

}

{
[Suite to run at].
HPT convert to TestNG
when suite anal]

5. ↓

Q] suite name changes ok??

→ < suite name = "suite" > → parallel = "tests" >

① Copy this and paste here
< test thread-count = "5" name = "Test1" >
< parameter name = "browser" value = "chrome" />
< classes >
< class name = "testng.A" />
< /class >
< /test >

① < test thread-count = "5" name = "Test2" >
< parameter name = "browser" value = "firefox" />
< classes >
< class name = "testng.A" />
< /classes >
< /test >
< /suite >

① (test) when </test> don't copy main </test>
② right < test > write right
< parameter name = "browser" name = "chrome" /> right.
③ < test > write Test1 or Test2 don't write Test1 or Test2
④ "suite" write right space then parallel = "tests" right

** Best assertion feature:- Hard Assert and soft Assert

```
Class A {  
    main() {  
        String expt = "Ankush";  
        String actT = "Ankush";  
        if(expt.equals(actT)) {  
            System.out.println("TC Pass");  
        }  
        else  
            System.out.println("TC fail");  
    }  
}
```

In above pgm, To verify expt and actT, We are using if ~~with~~ else conditional statement. So using if else statement length of script increases.

To reduce the length of script, we need to use Assert class. for verification.

Assert class contains static methods which are imported from org.junit.

- ① Assert.equals()
- ② Assert.notEquals()
- ③ Assert.assertTrue()
- ④ Assert.assertFalse()
- ⑤ Assert.fail() \Rightarrow [Intentionally we are failing TC]

① Assert.equals(): →

Used to verify expected and actual results, if both results are same then op is pass otherwise fail.

② Test

```
public void test()  
{  
    Reporter.log("running test method", true);  
    String actResult = "Hi";  
    String expResult = "Hi";  
    Assert.assertEquals(actResult, expResult);
```

here if actR and expR are same then op is pass.

if they are not equal then we will get exception in blue color.

② Assert.assertNotEquals(): →

Used to ^{verify} expected and actual results if they are not same then op is pass otherwise fail.

② Test

```
public void TC2(){
```

```
{  
    Reporter.log("running TC 2", true);  
    String actR = "Hi";  
    String expR = "H8";
```

```
    Assert.assertNotEquals(actR, expR);
```

```
}
```

```
}
```

③ Assert.assertEquals() →

It is used to verify conditions are true or false, if condition is true then OTP is pass otherwise fail.

Ex:- @Test

```
public void TC3() {  
    Reporter.log("running TC3", true);  
    boolean actResult = true;  
    Assert.assertEquals(actResult);  
}
```

⇒ For assertfalse reverse it

④ Assert.fail() →

It is used to intentionally fail the TC.

Ex:-

@Test

```
public void TC1() {  
    Reporter.log("TC1", true); // verification 1  
    Assert.fail();  
    System.out.println("Test is failed intentionally"); // verification 2  
}
```

@Test

```
public void TC2() {  
    Reporter.log("TC2", true);  
}
```

→ TC1 को multiple verifications किया गया।
उसकी fail होना तो multiple verifications
को पूरा करने की start, end
की verifications को terminate करता है।
[Test soft assert को multiple कहते हैं]

Disadvantages of assert class

If Test class contains multiple test cases if one of the Test cases contains multiple methods if one of the verification fails then entire remaining verifications also skipped. So to avoid this SoftAssert is invented.

SoftAssert :-

To overcome above drawback we need to use Softassert class.

It is class which contains all non-static methods which are useful for verification.

Ex:-

@Test

```
public void sample1() {
```

```
String str1 = "Hi";
```

```
String str2 = "Hello";
```

```
SoftAssert soft = new SoftAssert();
```

```
soft.assertEquals(str1, str2);
```

```
System.out.println("Hi, I am after failing 1st verification");
```

```
soft.assertAll();
```

```
}
```

@Test

```
public void sample2() {
```

```
Reporter.log("Hello", true);
```

```
}
```

** Difference b/w Hard Assert (Assert) and Soft Assert (Verify.) :-

OR

Difference b/w Assert and Verify:-

Assert

- ① If one of the verification is fail ~~is~~ belongs same TC/method then it will stop execution for next verifications
- ② Static methods of Assert class are used
- ③ No need to create object of Assert class
- ④ • No need to use assertAll() method.

Verify

- ① If one of the verification fails to same TC/method then it ^{will} continue execution for next verifications.
- ② Non-static methods of Assert class are used
- ③ we have to create object of Soft Assert class.
- ④ assertAll() are used. Compulsory

Method with Return type :-

```
Public class A {  
    public static int addition() {  
        int a = 10;  
        int b = 20;  
        int c = a+b;  
        System.out.println(c);  
        return c;  
    }  
}
```

```
Public static void main() {  
    int d = 50;  
    int e = addition();  
    System.out.println(e);  
}  
// 30  
// 80  
// 50
```

30 //

With POM and using TestNG :-

class.Kiteloginpage {

// step1 Declaration

@FindBy(xpath = " // input[@id = 'userid']") private WebElement UN;

@FindBy(xpath = " // input[@id = 'pwd']") private WebElement PWD;

@FindBy(xpath = " // input[@id = 'LoginBtn']") private WebElement LoginBtn;

// step2 Initialization

public Kiteloginpage (WebDriver driver) {

PageFactory.initElements (driver, this);

}

// Utilization

public void enterUN (String Username) {

UN.sendKeys (Username);

}

public void enterPWD (String password) {

PWD.sendKeys (password);

}

public void clickLoginBtn () {

LoginBtn.click ();

}

}

- ① Before class → open Browser
- ② Before method → open App
- ③ Test → Verify word
- ④ After method → close app
- ⑤ After class → close browser

Class Kitelogin2pageT

// Declaration

```
@FindBy(xpath = "____") private WebElement PIN;  
@FindBy(xpath = "____") private WebElement cntrbt;
```

// Initialization

```
public Kitelogin2pageT (WebDriver driver) {  
    PageFactory.initElements(driver, this);  
}
```

// Utilization

```
public void enterPIN (String pinvalue) {  
    PIN.sendKeys (pinvalue);  
}
```

```
public void clickcntrbt () {  
    cntrbt.click();  
}
```

```
}
```

Class KiteHomeT

// Declaration

@FindBy(xpath = " _____ ") private WebElement userID;

// Initialization

public KiteHomeT(WebDriver driver){

PageFactory.initElements(driver, this);

}

// Utilization

public String verifyUserID(){

String actID = userID.getText();

return actID;

}

}

Class KiteTest {

```
sheet sh;  
webdriver driver;  
Kitelogin1pageT login1;  
Kitelogin2pageT login2;  
Kitetomet home;
```

@BeforeClass

```
public void openBrowser () {
```

```
FileInputStream fis = new FileInputStream("ExcelPath");  
sh = WorkbookFactory.create(fis).getSheet("sheet1");  
System.setProperty("k", "v");  
driver = new ChromeDriver();  
driver.get("Kite.zerodha");  
// maximize  
// implicitly wait.
```

```
login1 = new Kitelogin1pageT(driver);  
login2 = new Kitelogin2pageT(driver);  
home = new Kitetomet(driver);
```

```
}
```

@BeforeMethod

```
public void openApp () {
```

```
String username = sh.getRow(0).getCell(0).getStringCellValue();  
login1.enterUN(username);  
String password = sh.getRow(0).getCell(1).getStringCellValue();  
login1.enterPWD(password);  
login1.clickLoginBtn();
```

String pinvalue = sh.getRow(0).getCell(2).getStringCellValue();
login2.enterPIN(pinvalue);
login2.clickentBtn();

@Test

```
public void verifyuserID(){}
```

```
Reporter.log("running Tc",true);
```

String actID = home.verifyuserID();

```
String expID = sh.getRow(0).getCell(0).getStringCellValue();
```

Value
Mandatory

method with
return

```
Assut.assertEquals(actID,expID,"failed Both results are  
diff");  
}
```

@AfterMethod

```
public void logout(){}
```

```
Reporter.log(" logout from App",true);
```

```
}
```

@AfterClass

```
public void closebrowser(){}
```

```
Reporter.log("close browser",true);
```

```
}
```

Difference b/w Junit and TestNG.

Junit

- ① It doesn't support to parallel execution
- ② It does not support advanced annotations
- ③ The dependency tests are missing in Junit
- ④ Grouping tests together is not possible

TestNG

- ① It supports to parallel execution
- ② It supports advanced annotations
- ③ Dependency tests are present in TestNG
- ④ Grouping of tests is possible.

~~④ Pom with Pagefactory using Base Utility :-~~

Create Maven project

What is Maven project ?

⇒ It is build testing tool

What is build ?

⇒ compiled and compressed file is called as build.

What is Library ?

⇒ collection of Jars is called as Library

Build testing tools :-

① Ant → previously used

② Maven → currently using.

Maven usage in Automation :-

Using Maven, No need to add externally jars and library
Because we are adding dependencies from maven repository
these dependencies automatically adds updated jars / libraries

Maven goals / Maven life cycle :-

① Clean :- It is used to uninstall previous build and install new build.

② Install :- It helps to download all files required for project into Local machine.

③ Compile :- We need to add compile plugin (∵ JDK is not compatible we need 1.8 version)

④ Test :- It is responsible to execute the script.
[surefire plugin]

Difference b/w Java project and maven project

Simple project

- ① we can't add dependency as simple.
- ② If latest version comes then we have to add latest version
- ③ Possibility of merging project is high.

Mvn project

- ① we can add dependency as simple.
- ② simply add dependencies automatically latest version will be updated.
- ③ Here is no possibility of merging project.

Core selenium

Core selenium is tool oriented.

Advanced selenium

Advanced selenium is process oriented.

ex:- Maven / Ant / Gradle

Q] How to add dependencies in Maven ??

⇒ 1) ~~1st~~ go in pom.xml file (double click on it because inside this we can add the dependencies)

2) After ~~version~~ line simply write i.e. dependencies

(dep control + space)

Provide it add dependencies from maven Repository

Ex:- <Project>

<Version>

<dependencies>

</dependencies>

</version>

</project>

Here we can add dependency (4 line of code)

① selenium Java (4.1.2)

② Apache POI common (5.2.2) ←

③ Apache POI-OOXML (5.2.2) ^{schema} → [Same version at
प्रारंभिक रूप से]

④ WebDriver Manager (5.2.0) ^{catalog}

⑥ Testing.

→ जोड़ना फ्रेट.

* Work of WebDriver Manager →

Here ~~in~~ In maven, driverexecutable files are not added, for ex. Instead of taking chrome, firefpx, Safari, opera.. we need to add WebDriver manager dependency so that automatically all the jars of driverexecutable files will added.

* Difference b/w Sel3 and Sel4 →

Selenium 3

- ① JSON wire protocol.
- ② minimize is not there
- ③ implicitly wait

Selenium 4

- ① W3C protocol
- ② Minimize is there
- ③ Duration • OF seconds

- * Work of pom.xml : → ① It is project object model in which we can add dependencies
② useful to add all related dependencies

* How to update maven project ?? ~~to my project~~
~~red color in standard~~

- Right click on project → maven → update → forcesomething
→ automatically it will update.

Suggestion : → Don't play with pom.xml file

* maven project structure : →

src/main/java

src/main/resources

src/test/java

src/test/resources

[here we can create package
we can create class
we can write the code]

* POM with PageFactory using TestNG-Base Utility Class

① src/main/java.

↓

plg → pom classes.

↳

copy paste all 3 classes here

Control + C.

② src/test/java.

↳ plg → Base

Class Based

Public WebDriver driver;

Public void initializeBrowser () {

System.setProperty ("k", "v");
driver = new ChromeDriver();
d.set ("kite");

Maximize();

ImplicitWait();

}

}

=====

③ src/test/java.

↳ plg Utility

[Return type concept]

Class Utility of

Public static ~~String~~ void getTD (int rowIndex, int colIndex) throws

FIS fis = new FIS ("Excelpath");

Sheet sh = workbook.create(fis).getSheet ("Sheet1");

String data = sh.getRow(rowIndex).getCell (cellIndex).getStr (cellIndex);

return data;

④

src/test/java.

↳ Pls Testclass

Class Test extends Baseclass {

KiteLoginPage login1;

KiteLoginPage login2;

KiteHomepage home;

@Before class

public void openBrowser () {

initializeBrowser(); // Non static call from Non static

login1 = new KiteLoginPage (driver);

login2 = new KiteLoginPage (driver);

home = new KiteHomepage (driver);

}

@Before method

public void LogintoApp () {

login1.enterusername (Utilityclass.getTD (0,0));

// sys0 (Utility.setTD (0,0)); // class name & method name . call by

login1.enterpwd (Utilityclass.getTD (0,1)); // DPG458

login1.clickLoginBtn ();

login2.enterpin (Utilityclass.getTD (0,2));

login2.clickCntBth ();

}

@Test

public void verifyUserID () {

Reporterlog ("Running verify UserID", true);

String actID = home.verifyUserID ();

String expID = Utilityclass.getTD (0,0);

Assert.assertEquals (expID, actID, "both are diff & this is fail");

}

@After method

public void logout () {

Reporterlog ("Logout", true);

}

@After class public void closeBrowser () {

卷之三

卷之三

Jenkins

ROI [Return of Investment]

2012 → Task scheduler

2013 | 2014 ⇒ Jenkins ~~git~~

JENKINS :- It is CI/CD tool. (contains Intg, conts Deploy)

Jenkins will be used to automate build creation and build testing and build deployment process so it is called as continuous deployment tool.

Jenkins Usage in Automations

In Automation, Jenkins will be used in continuous integration integration process. It means Jenkins will be always monitor the framework in Git location, automatically creates a build and provides email notification if any changes had happened in the build.

Jenkins can also be used to execute all the testscripts whenever new build arrives to testing Environment.

Jenkins provides 3-levels of Execution

- ⇒ ① on-demand
- ② on-schedule
- ③ poll-SCM (source code management)

① on demand :- Based on the customer demand, login to Jenkins and start the Execution.

② on-schedule :- Based on the schedule time, which is given in the configure area, Jenkins automatically triggers build.

③ poll SCM :- Poll means Continuously Monitor the framework/app in git location, it automatically starts the Execution whenever any changes happen in the framework or testing server.

Note: → Whenever Jenkins starts the Job Execution it will connect to git and download the entire framework (build) into local system and test the build by taking help of Maven and execute all the test scripts and sends out an email.

Jenkins Plugins: →

- a) Maven Integration plugin
- b) GitHub Integration plugin.

Q] How to create project/Job in Jenkins ??

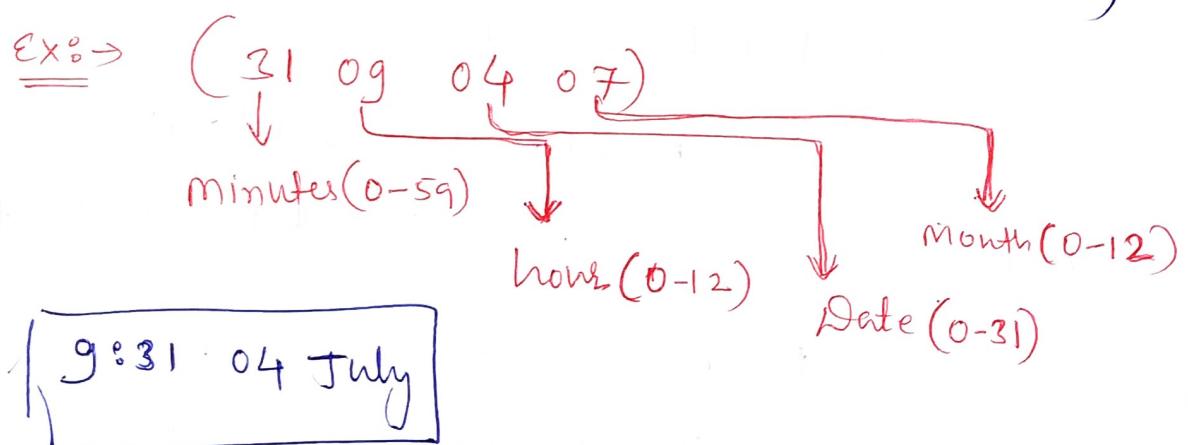
- ⇒ 1) Login to Jenkins
- 2) Click on new item, select maven project and click on OK.

③

Q] How to configure schedule

- ⇒ ① Login to Jenkins
- ② Go to Configure file of the Job
- ③ Scroll down and select build periodically checkbox.
- ④ Provide the time wrt. (min, hour, date, month)

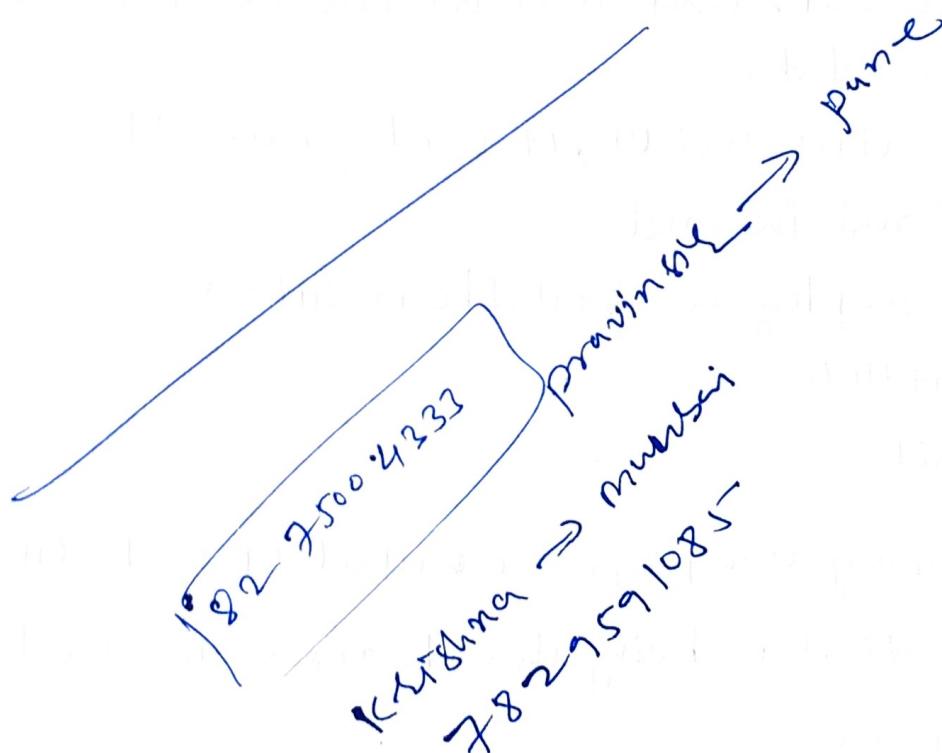
⑤



Q] How to configure to email Q2

- ① Go to Jenkins project configure file
- ② Select email notification checkbox at the end of the stage.
- ③ Provide recipients details.

Note: → In order to configure email notification, we should provide company SMTP credentials in global configuration file.



★ Source control tool / Configuration management tool /
Version control tool / Source code management tool :-

- ① GitHub
- ② SVN (Subversion)
- ③ CVS (Concurrent version system)
- ④ perforce

GitHub → It is cloud based decentralised repository where we should create an acct with GitHub community in order to use it.

It is centralised repo in which source code of framework can be stored. and there we can create our own rep. to hold the data.

It provides web UI, it can be accessible anywhere in the world through internet.

There are 2-plugins available in Git:-

- ① GitHub
- ② Git

(Based on money you pay, you will get space 10 GB, 10TB etc) that is being shared across the world.

Advantages of GitHub :-

- 1) It is cloud based decentralized repo.
- 2) No need to invest money for physical hardware
- 3) No need of any admin to maintain the storage
- 4) It provides Web Interface
- 5) It can be easily accessible anywhere in the world through internet
- 6) It provides security and Backup facility.

Git → It is client plugin which should be installed in local system which is used to access github over the net (internet).

Without git plugin we can't communicate with github.

There are many plugins available in market:— github.

- ① E-Git
- ② GitBash (cmd prompt)
- ③ Git-Desktop (cmd prompt)

~~BDD Cucumber~~

~~BDD~~ → Behaviors driven development

TDD

→ Test driven Development.

Using TestNG, whatever we are writing the code it is understandable by technical persons only [user/tester]

But in case of BDD cucumber the code which is written is understandable by even non-teaching peoples also. hence BDD is userfriendly than TestNG.

TDD

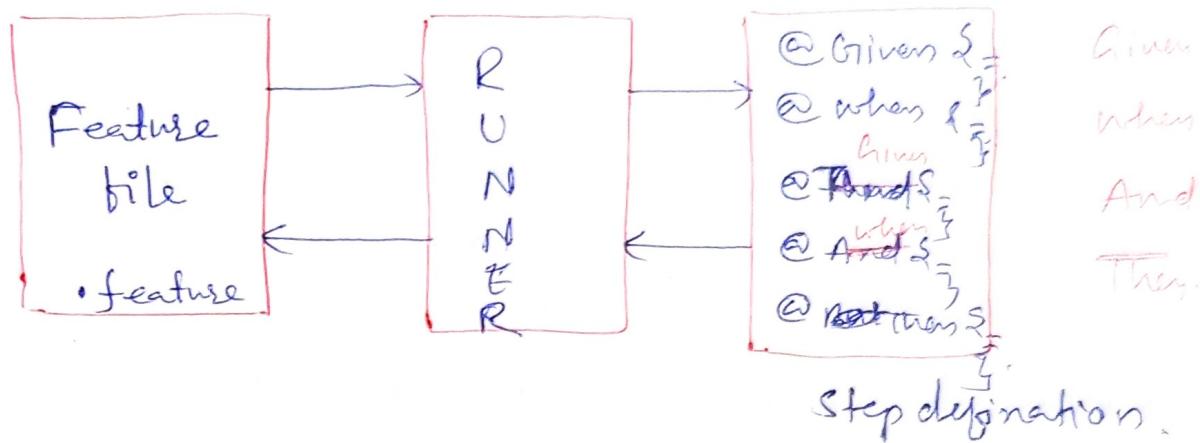
```
Class A
@Test
P v Test() {
R. L. ("Hi", true);
}
}
```

Now if we want to write code inside BDD cucumber we need 3 - resources / 2 things.

Cucumber resources →

- ① Feature file → .feature
- ② Step definition →
- ③ Runner class : - [Test Runner]

Execution Diagram :-



Flow :-

Feature: Google search TextBox verification

Scenario: verify Text is accepting data.

Given: User is on Homepage (precondition)

When: User can enter vw, pass

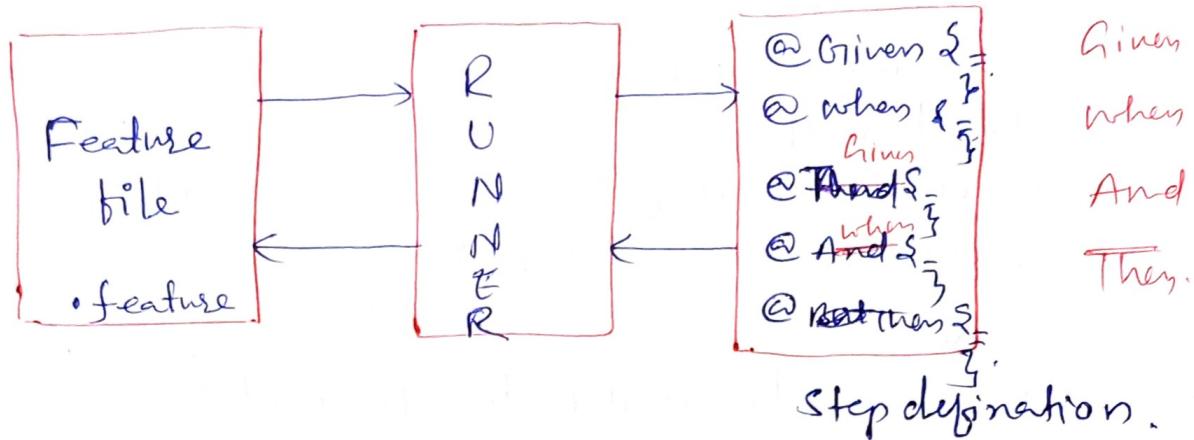
And: User can click on Login Btr.

Then: Homepage must be displayed.

In Featurefile we will maintain Gherkin Language.

[
It is like normal English language anyone can understand this.]

Execution Diagram :-



Flow :-

Feature: Google search TextBox verification

Scenario: verify Text is accepting data.

Given: User is on Homepage (precondition)

When: User can enter UN, pass

And: User can click on Login btn.

Then: Homepage must be displayed.

In Featurefile we will maintain Gherkin Language.

[It is like normal English language anyone can understand this.]

~~E-2~~ ① Create new ^{Maven} project in Eclipse and add dependencies

~~steps~~

- cucumberJava (7.6)
- Selenium - Java (4.3.0)
- Junit (4.13.2)
- cucumberJunit (7.5.0) (Installメント)
- webdrivermanager (catatum)



Once it is done plz update the project ??

⇒ Right click on project ⇒ Maven ⇒ ~~Force update~~ =)

Force ⇒ It will update successfully.

② Download cucumber plugin from Eclipse market place.

⇒ Go to Help ⇒ Eclipse marketplace ⇒ search for Cucumber



I accept



Update the project.

③ Create .feature file under Features folder under src/test/resources



→ Features



→ ~~Acti~~.ActiLogin.feature

④ Inside feature file write.

Feature: Feature to test Login functionality

Scenario: check login is successful with valid credentials

Given: User is on login page

When: User enters and press

And: Click on Login btn

Then: User is navigated to the home page.

⑤ Create stepdefinition under ~~step definitions~~ ~~src/test/java~~



Folder → StepDefinitions



file → Activesteps.

[here provide the stepdefinition. we need to paste the O/P of feature file from @ Given]

Ex: → class: Loginsteps



```
@Given("user is on login page")
P v methodname()
{
    sys(" ");
```

@When()

@

!

!

!

@Then

```
P v methodname()
{
    sys(" ");
}
```

!

import ~~GivenThen~~

io.cucumber.Given;

io.cucumber.*;

All will automatically imported

⑥ Create runner class under ~~src~~ under StepDefinition Pkg

If we want to run one feature file we can run directly right click ⇒ Run as Cucumber. But If we want to run multiple feature files or scenarios we have to run the Runner class.

Ex :- ~~Class Activator~~

Runner class → It is responsible to run step definition and feature file.

@RunWith(Cucumber.class)

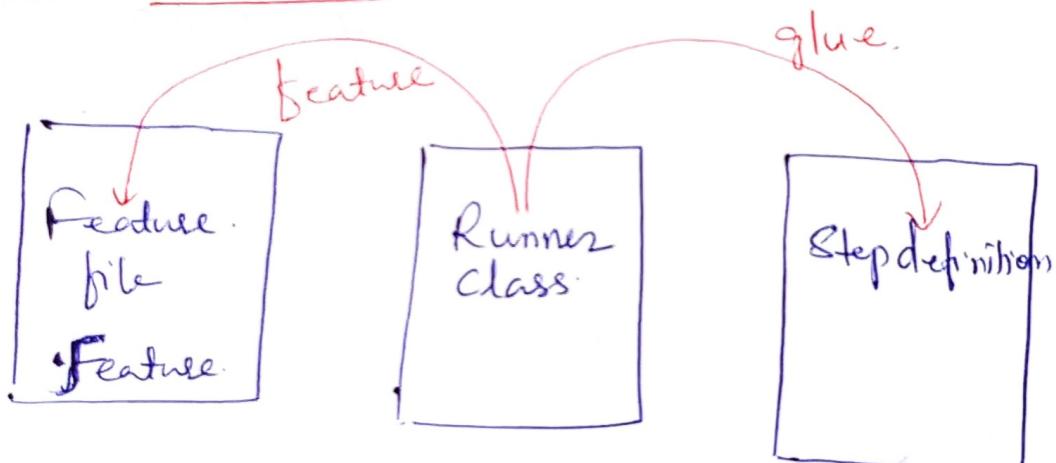
@ CucumberOptions(features = "featurefilelocation", glue = {"stepDefinition"})

class Activator { } }.

① [right click → properties → location
→ from src to end Copy And]

Name should be same package of stepDefn

Execution Diagram



C-3

Report generation in BDD cucumber

We can generate reports in multiple format. For Ex:-
HTML, JSON, XML format.

But to get report in different format we need to use plugin

Plugin = `{"pretty": true, "html: target/html/reports.html"}`

Procedure :- It is the format

1) Create folder HTML under target Folder

under HTML create file "html-report"

2) Give the path of this file \Rightarrow right click to html-reports \Rightarrow

Properties \Rightarrow location \Rightarrow from target copy all and paste in Runner class

Ex:-

Runner class:-

`@RunWith(Cucumber.class)`

`@CucumberOptions(features = "featurepath", glue = {"stepdefinition"},`
`Plugin = {"pretty": true, "html: target/html/report.html", "json: target/`
`jsonreports/reports.json", "junit: target/target/report.xml"}`

Actitime using Cucumber

① create feature file under src/test/resources

FEATURE

→ Actitime.feature

Feature: check Login Functionality of Actitime

Scenario: Verify User is login successfully with valid cred

Given Browser is open

And User is on login page

When User Enter UN and pwd.

And Click on Login btn

Then User should navigate to Homepage.

↓ Run this and copy & paste into stepdefinition

② create stepdefinition under src/test/java

StepDefinition

→ class ActitimeStepD

Class ActitimeStepD {

webDriver driver = null;

@ Given ("Browser is open")

P v. B is open () {

System.setProperty("webdriver.chrome.driver",

driver = new CD();

implicitlyWait(20);

}

@ And ("User is on Login page")

P v. User is on login page () {

driver.get("url");

}

@ When ("User Enter UN & pwd")

P v. User Enter UN & pwd () {

```
dr.findElement(By.xpath("//input[@name='username']")).sendKeys("admin");
dr.findElement(By.xpath("//input[@name='password']")).sendKeys("manager");
}
```

@And ("click on login button")

```
PV.clickOnLoginBtn();
```

```
dr.findElement(By.xpath("//input[@type='button']")).click();
```

```
}
```

@Then ("User should Navigate to Homepage")

```
PV.userShouldNavigate();
```

```
String expD = "sep29.11";
```

~~Story:~~ String actD = dr.findElement(By.xpath("//xpathofdate")).getText();

```
if (expD.equals(actD)) {
```

```
System.out.println("Navigate to HPP, Pass");
```

```
}
```

```
else {
```

```
System.out.println("Fail");
```

```
}
```

(3) Runner class :-

PackageName

@RunWith(Cucumber.class)

@CucumberOptions(features = "Path of featurefile", glue = {"StepDefn"},
plugin = {"pretty", html:target\html\report.html"}}

```
public class ActimeRunner {
```

```
}
```

Parameterization :- It is process in which we are accessing test data from external source.

Ex:- Examples:

Examples :- It is the Placeholder element through which we can pass the parameter.

Ex:- Examples:

|UN|PWD|

|admin|manager|

Q] How to run feature file multiple times with different sets of data ??

→ Step 1 :- Feature file

Feature: Check login functionality of Achitme

Scenario outline: Verify user is login successful with V-D

Given Browser is open on chrome

And User is on Login page

When User enters <UN> and <PWD>

And User click on login btn

Then User should navigate to home page

|

|

|

Examples:

|UN|PWD|

|admin|manager|

→ ~~Get black text~~

→ ~~Get white text~~

Examples - using space

→ ~~Get text next line in front~~

StepDefn:-

```
class parasteps {
```

```
    WebDriver driver = null;
```

```
@Given ("Browser is open on chrome")
```

```
PV Browser is open on chrome () {
```

```
System.setProperty("k", "v");
```

```
driver = new CD();
```

```
} implicitlyWait();
```

```
@And ("User is on login page")
```

```
PV User is on login () {
```

```
driver.get("https://demo.actitime.com");
```

```
}
```

```
@When ("User enters (.*) and (.*) $")
```

```
PV User enters admin manager (String UN, String PWD) {
```

```
dr.findElement("xpath of UN").sendKeys(UN);
```

```
dr.findElement("xpath of pwd").sendKeys(PWD);
```

```
}
```

```
@And ("User clicks on login btn")
```

```
PV User clicks on login () {
```

```
dr.findElement(By.xpath("xpath of click"));
```

```
}
```

```
@Then ("User should navigate to home page")
```

```
PV User should navigate to home page () {
```

```
String expD = "Aug 29-sep 04, 2022";
```

```
String actD = dr.findElement(By.xpath("____")).getText();
```

```
if (actD.equals(expD)) {
```

```
    System.out.println("User navigate to homepage" + tc1pass);
```

```
    else
```

```
    System.out.println("TC is fail");
```

```
}
```

Rumer class :-

@RunWith(Cucumber.class)

```
@CucumberOptions(features = "src/test/features", glue = {"step"},  
public class StepRunner {    plugin = {"pretty", "html:target\\\"})  
    cucumber  
}
```

Diagram illustrating the relationship between expressions and regular expressions:

- A red circle labeled **(*)** is connected by an arrow to a red bracket containing the text **variables or let F605n**.
- A blue line labeled **Expression** with a downward arrow is connected to a red bracket containing the text **multiple times symbol UN & prod change on ET Regular expression automatic UN & prod ET ET**.

6-6

DOM classes Designing.

Step 1 → create feature file.

↳ kite.feature

Feature: Verify Login functionality

Scenario Outline: Verify kite.zerodha login successful with VD

Given Browser is open in chrome

And User is on ~~http://~~ login page

Then

When User enters <UN> & <PWD>

And Click on login btn

Then User navigate to login page and enter <PIN>

And Click on continue btn

Then Verify login successful with ~~http://~~ <userId>

!

Examples:

| UN | PWD | PIN | userId |

| Dp4458 | Amol2144 | 171992 | Dp4458

~~DOM~~ (src/main/java)

~~Class: DOM12~~

Under src/main/java:-

Class POM1 {

```
@FindBy(id = "userid") private WebElement userid; UserID  
@FindBy(id = "password") private WebElement password; password  
@FindBy(xpath = " _____ ") private WebElement loginbtn;
```

Public POM1 (webdriver) {

```
    PageFactory.initElements(driver, this);  
}
```

```
Public void enteruserid (String userid) {  
    userid.sendKeys(userid);  
}
```

```
Public void enterpassword (String pwd) {  
    password.entersendKeys(pwd);  
}
```

```
Public void clickLoginBtn () {
```

```
    LoginBTN.click();  
}
```

```
}
```

Class POM2 {

```
@FindBy(id = "pin") private WebElement PIN;  
@FindBy(xpath = " _____ ") private WebElement CNTBTN;
```

Public POM2 (webdriver) {

```
    PageFactory.initElements(driver, this);  
}
```

```
Public void enterpin (String pin) {  
    PIN.sendKeys(pin);  
}
```

```
public void clickBtn() {
    ENT_BTN.click();
}
```

class Home{

```
@FindBy(xpath = " //span[@class = 'user']") private WebElement verifyUser();
```

```
public Home(WebDriver driver) {
```

```
    PageFactory.initElements(driver, this);
```

```
}
```

```
public String verifyUserID() {
```

```
    String actR = verifyUser.getText();
```

```
    return actR;
```

```
},
```

```
}
```

Class Kitestep { [src/test/java] }

public webd⁸.driver=null;

POM1 login1;

POM2 login2;

Home home;

@ Given ("Browser is open on chrome")

P V m() {

System.out.println("K", "V");

driver=new webd⁸;

driver.manage.time.implicitWait.

}

@ And ("User is on login page")

P V m() {

driver.get("zerodha");

}

@ When ("A user enters (*) and (*) \$")

P V m (String UN, String PWD) {

login1=new POM1(driver);

login1.enterUserId.(UN);

login1.enterPassword.(PWD);

};

@ And ("click on login btn")

P V m() {

login1.clickLoginBtn();

};

@ Then ("A user navigates to login2 page enter(*) \$")

P V m (String int pin)

String PIN = Putterer.sendKeys(pin);

login2=new POM2(driver);

login2.enterPIN();

@ And ("check on cont Btu")

PV m() {

}
down2::checkBtu();

@ Then ("^ verify world: successful with (**) \$")

PV m(string userID) {

home = new Home(driver);

string expr = userID;

string actR = home.verifyID();

if (_____ -> {

synd

}{

}{

}{

}{

Testrunner

[As it is]

BDD Cucumber Questions and Answers:-

① What is BDD cucumber?

⇒ It is the behaviour driven development framework, the Speciality of BDD cucumber is even Non-technical peoples also understand the features and scenarios of application.

② What are Cucumber resources?

⇒ ① Feature file (which have extension as .feature)
② Step Definition (In which we can write actual script)
③ Runner class (which is responsible to run step definition and featurefile simultaneously)

③ What is use of features and glue?

⇒ features is used to provide path of featurefile while glue is responsible to provide path name of stepdefinition folder.

④ Which annotation is responsible to run the Testrunner?

⇒ @RunWith (Cucumber.class)

⑤ Which are the ways we generate Reports in ~~Test~~ BDD?

⇒ we can generate 1) HTML 2) JSON, 3) XML reports by providing path of target folder in "pretty" format.

⑥ What is parameterization?

⇒ The process of providing data from Cucumber element i.e Examples is called as parameterization

Ex:- |UN| prod|
|admin| manager|

Q] What is Data Driven Testing?

⇒ The process of fetching data from 3rd party tools/external resources is called as Data Driven Testing.

Q] What is POM? Advantages of POM?

⇒ Refer my selenium notes.

Q] Where to create POM classes in Cucumber Maven project?

⇒ In src/main/java.

Q] Which expression we use in BDD?

⇒ ^ \$

→ Inside it we can write anything.

Q] What are Tags

⇒ Tags are the names which are given in Feature file for feature identification purpose.

Feature file
~~runner class~~

@smoke @regression @important

Feature: verify login functionality

@smoke

Scenario: sample1

Given
when
Then
And

@regression

Scenario: s2

Given
when
Then
And

②

Q] What is Scenario Outline?

⇒ It is a Gherkin Element through which we can provide the set of test data.

Q] What are Hooks?

⇒ These are block of codes which runs before and after each scenario.

Ex:-

Scenario Hooks - runs before and after each scenario.

Step Hooks - runs before and after each step.

Conditional Hooks - runs according to tag.

Q] Why to use Hooks?

⇒ To manage setup.

Q] What is Background?

⇒ These are group of steps that are common to all scenarios in feature and we can define it once in Background.

Q] Why to use Background?

⇒ 1) To avoid repeating the common steps in every scenario
2) for better readability purpose and maintenance.

Q] When to use Background?

⇒ whenever there are common repeating steps in a features.

Difference b/w Sel3 & Sel4 :-

- | | |
|--|-------------------------|
| 1) JSON wire protocol | 1) W3C protocol. |
| 2) minize() is not there | 2) minime is there |
| 3) implicitly wait (20, TimeUnit, seconds) | 3) (Duration::FSECONDS) |

* How to configure Maven to windows ??

~~Rough~~

Install

① In Browser enter <https://maven.apache.org/download.cgi>
OR

Enter maven.apache in Browser

②

Click on Binary zip Archive [apache-maven-3.9.2-bin.zip]
↓
Download [Extract it].

③ Check in program file [ie in Local c: disk]

i.e. Thispc > Local Disk (C) > programs > maven > apache-maven-3.8.4

- bin
- boot
- conf
- lib
- license
- NOTICE
- README.txt

④ Search for Edit system Environment Variables

↓

Environment variables

System Variables: →

New sys^m variable: —

Variable Name: ①

Variable Value: ②



② ↴

Sys^m variables: —

Variable

Path

PathNext

make
blue
by clicking

↴

↴

Edit Env Variables: → Add mvn Home Directory in path

| C:\Program Files (86)\common files\oracle\Java\j2se\path

↓ make it as blue

New

Edit

OK

* Click on New button then edit environment variable window

C:\Program Files

Maven - Installation process :-

Step 1

- ① In Browser enter <https://maven.apache.org/download.cgi>
OR

Enter maven.apache in Browser



Click on Binary zip Archive [apache-maven-3.9.2-bin.zip]



Download [Extract it]

- ② check in program files [i.e in Local C:Disk]

i.e Thispc > Local disk(C) > program files > maven > apache-mvn-3.9.2

bin

bat

conf

lib

licns

NOTICE

README.txt



 [प्रोजेक्ट में बना गया
31 नवम्बर, 2021]

पोर्टल
पोर्टल

Port

Step 2 Add MAVEN_HOME System Variable

- 1) Open the start menu and search for Environmental Variable
- 2) click the Edit the system env^t variables
↓

Environment variables ↵

- 3) Click New button under the system variables section to add a new system ~~Environment~~ Variable

System variables:-

New Edit Delete

[Don't click on anyone]

New ↵ Edit Delete

OK ↵

- 4) Enter MAVEN_HOME as variable name and as a Variable value c:\psemfiles\maven\apache-mvn-2.9.2

OK ↵

Step 3 Add MAVEN_HOME Directory in Path Variable



System Variables :-

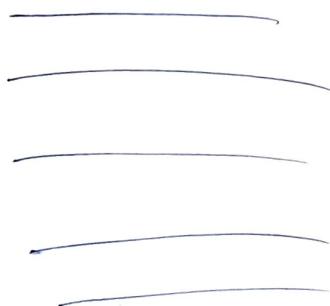
Variable

Path

Value

C:\Pgm\htest\com\com1\java\javac\bin

- 1) click New Btn in the Edit Environment variable window



- 2) Enter $\%MAVEN_HOME%\bin$ in the new field. click "OK" to save changes to the Path variable

File



$\%MAVEN_HOME%\bin$

3) finally click on OK ↴

Step 4 Verify Maven Installation



Windows + R \Rightarrow enter cmd ↴



mvn -version ↴

=====
=====
=====
=====
====

=====
=====
=====
=====
====

=====
=====
=====
=====
====

=====
=====
=====
=====
====

C:\Users\larkova>.



How to configure TestNG

Click on help



Eclipse marketplace



Search for testng (type testng and click on search (R))



Testng for eclipse will come, so click on install



Click on confirm



click on checkbox and finish.

Q]. How to add to Eclipse ??

- ⇒ ① Right click on project where u have to configure
② Build path ⇒ Add library ⇒ Finish.

Another method to Configure TestNG for Eclipse:-

In google search for `testng.org`

↓
Go in download (2nd option in tabs)

↓
scroll down and see "Install from Update site"

copy:- `https://testng.org/testng-eclipse-update-site.`

↑ while paste,
remove .

Go in Help ⇒

Install new software

↓

Add ← click here

works with:-

don't write.

↓

Add Repository:-

Name :- **testNG**

← write this

Location:- `https://` ← paste here [remove.]

↓

Add

↓

Next

↓

Finish.

प्रीभयी प्रिमकरांत अनुप विजवांगी नंतर शेपभांत्रा अंते तेंदांन
हे वर्णांतक कविता आहे.

अलगद-गालगद उघडून घेते ती द्वितीये हार कृदान्ति,
हृदय जावते स्पराने अंग तिचे अंगार कृदान्ति. } $\Rightarrow 2$

✓ साडिवरसी भाज पार ती वारत होती कोशी-कोरी
कुळावाचुन अपुरा पुला साधगेचा शुंगार कृदान्ति. } $\Rightarrow 2$

✓ भीरभांत मानांक ती दिवसाळुकव्या घेतच जाणि
गिशिंगांद्यातील रात्राणी तिचे जावडतो भींधार कृदान्ति } $\Rightarrow 2$

✓ नाही बोलली भाजभावरती, जाणी बोलली जातांका
कुराव्यांने रोश्युन धरला कुंठातील उकार कृदान्ति } $\Rightarrow 2$

✓ सतबद्य होते शब्द तिचे भाज कोरडे होते काहर झोठ
आणि येप्यापुरी वरयुन माळे डोऱ्यांचे माहार कृदान्ति } $\Rightarrow 2$

Section-2 [selenium]

Ankush

Date :- 27/07/2022

① LB :-

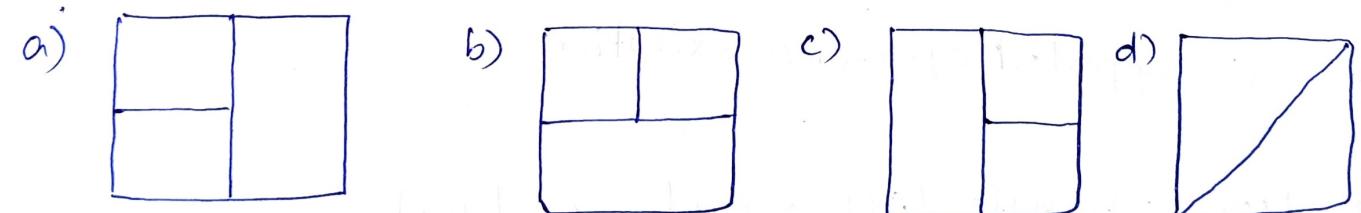
- ① what is LB/WD ??
- ② what is the difference b/w static DD and Dynamic DD.
- ③ what are methods in select class ?? [selection methods, Deselection methods and operational methods explain all]
- ④ Tagname of given List Box is not at all select But we are trying to handle Listbox using Select class then which exception we will get ??
⇒ Unexpected tagname Exception.
[Ex:- we are waiting for girlfriend suddenly wife came]
- ⑤ How to create multiselectable WD ??
- ⑥ Can we use Deselection methods for singleselectable DD ??
- ⑦ How to print all options in the Listbox ??
- ⑧ What happens if we try deselect option in singleselectable LB ??
⇒ Unsupported operation Exception
- ⑨ How to handle ListBox Explain in detail ?

② Autosuggestions :-

- ① How to handle Autosuggestions explain in detail .?
- ② How to inspect for Autosuggestions ?
- ③ How to click on particular element in Autosuggestion ??

③ Iframe:-

- ① By using which tagname Iframes are created?
 - ② How to see how many frames are present on webpage?
 - ③ By default where is the control of our selenium?
 - ④ How to switch from mainpage to frame?
 - ⑤ How to switch from any child to main webpage?
 - ⑥ While switching to frame if we use class then which exception we will get?
⇒ "No such frame exception".
 - ⑦ What is work of parentframe ()?
 - ⑧ What is work of default content ()?
 - ⑨ Can we switch directly from one frame to another frame in Automation?
 - ⑩ What is frame/iframe?
-



④ Popups:-

- ① What are popups??
- ② Can we automate window and Browser popup using ~~vinay~~ Selenium WebDriver??
- ③ Why popups will come in Automation?
- ④ Can we say Hidden division popup as HTML popup??

- ⑤ Can we say Alert popup as Javascript popup??
- ⑥ By using which tagname Hidden-Division popups are generated?
- ⑦ What are characteristics of HTML popup?
- ⑧ How to handle HTML popup?
- ⑨ What is Alert ?? class or Interface ??
- ⑩ What are characteristics of Alert popup?
- ⑪ What are the methods in Alert Interface? What is use of it??
- ⑫ For which type of Applications Notification popups will come?
- ⑬ Which is the Browser setting class for chrome Browser?
- ⑭ How to handle Notification popup?
- ⑮ What is chromeoptions ??

Automation Testing

Java syllabus :-

- ① why we are using Java to develop the application ?
- ② what is Java ? what are features of it ? what are pillars of Java ?
- ③ what are Translators in Java ?
- ④ How Java is platform independent ?
- ⑤ why os is called ~~not~~ as platform ?
- ⑥ what are variables ? and its types ?
- ⑦ what are data types and its types ?
- ⑧ what is size of primitive data types ?
- ⑨ Difference b/w primitive datatypes and Non-primitive data types ?
- ⑩ what is class ?
- ⑪ what is an object ?
- ⑫ what are methods ? what are types of methods ?
- ⑬ what is constructor ? what are its types ?
- ⑭ what is the difference b/w Method and constructor ?
- ⑮ what is use of parameterized constructor ?
- ⑯ Can we overload constructor ?
- ⑰ Can we override constructor ?
- ⑱ How constructor chaining is possible ?
- ⑲ what is the syntax of main method ?
- ⑳ why main() is static ?
- ㉑ what is static Initializer block ?
- ㉒ Can we overload main() ?
- ㉓ Can we override main() ?
- ㉔ what are keywords in Java, what is work of keyword ?
- ㉕ what are Identifiers .
- ㉖ what is use of control statements ?
- ㉗ why loops are used in Java ?
- ㉘ what are oops principles ? what is use of it ?

- 29 - What is mean by Inheritance?
- 30 - Can single class exhibits property of Singlelevel Inheritance?
- 31 - What is singlelevel, multilevel, hierarchical Inheritance?
- 32 - What is the use of extends keyword?
- 33 - What is superclass and subclass?
- 34 - What is multiple Inheritance?
- 35 - Can we achieve multiple Inheritance in Java using class?
- 36 - How we can solve DAP?
- 37 - How DAP occurs?
- 38 - Who is responsible to create DAP?
- 39 - What is supermost class in Java?
- 40 - What is this and super keyword?
- 41 - What are Access Specifiers?
- 42 - What is abstract class? Concrete class?
- 43 - Can we achieve 100% abstraction in Java?
- 44 - What is Interface?
- 45 - Can we achieve 100% abstraction using Interface?
- 46 - What is method overloading?
- 47 - What is method overriding?
- 48 - Difference b/w method overloading and overriding.
- 49 - Explain JVM architecture.
- 50 - What is upcasting?
- 51 - What is implicit and explicit casting?
- 52 - When we get class cast exception?
- 53 - What is exception? How to handle it?
- 54 - Which is supermost class for Exception classes?

- 55 - Can we use multiple try blocks in same class?
- 56 - What is finally block?
- 57 - What is mean by checked Exception?
- 58 - What are unchecked Exceptions?
- 59 - What is difference b/w throw and throws?
- 60 - Tell me any five exceptions in Java?
- 61 - What is string class? What are features of it?
- 62 - Explain in detail all methods of string class?
- 63 - What is difference b/w string, Stringbuffer, Stringbuilder?
- 64 - What is Array? What are features of it?
- 65 - What is single dimensional and multidimensional Array?
- 66 - What is collection? Explain all implementation classes in Collection
- 67 - What are wrapper classes?
- 68 - What is cursor? What are 3-types?
- 69 - What are Disadvantages of manual Testing?
- 70 - What are advantages of Automation Testing?
- 71 - What is selenium? What are its advantages?
- 72 - What are disadvantages of selenium?
- 73 - What are selenium flavours?
- 74 - Explain architecture of selenium webDR?
- 75 - Explain 14-methods in WebDriver Interface?
- 76 - What is web element? What are methods in it?
- 77 - Which class is used for set the size of webpage?
- 78 - Which class is used to change the position of webpage?
- 79 - What is tagname, attribute and text?
- 80 - What are locators? In which class does it presents?
- 81 - Which one fast locator among all?
- 82 - Why we are not using CSS Selector in all scenarios?
- 83 - What is xpath?
- 84 - What is difference b/w Absolute and Relative xpath?
- 85 - When we use xpath by descendant.
- 86 - What is keys class? What are its disadvantages?

① Robot class:-

- 1) what is use of it?
- 2) In which pkg does it belongs?
- 3) what are advantages of Robot class?

② List Box / Drop Down:-

- 1) How to handle Listbox?
- 2) What is select class and what are methods in it?
- 3) If we use deselect methods for singliselectable class then what happens?

③ Screenshot:-

- 1) How to take screenshot?
- 2) Write code in notepad to take screenshot?
- 3) What is Takescreenshot?

④ Synchronization:-

- 1) What is Java wait?
- 2) What are selenium waits?
- 3) What is Difference b/w Implicit and Explicit wait?
- 4) What is fluent wait.

⑤ Scroll-down - up ↕

- 1) How we can scroll-down in Automation?
- 2) What is JavascriptExecutor?
- 3) What is executeScript?

⑥ 2frames:-

- 1) How to identify our page is divided into frames?
- 2) what are i-frames? what are methods are these to handle i-frame?

8)

⑦ Popups:-

- 1) what are popups?
- 2) How to handle childbrowser popup?
- 3) what is mean by window handling?
- 4) How to handle Notification popup?
- 5) How to handle Hidden-division popup?
- 6) what is Alert?
- 7) How to handle Alert popup?

9)

⑧ How to handle multiple Elements? what is use of findelements method?

⑨ Webtable:-

- 1) what is webtable? how it is created?
- 2) How to handle webtable?

⑩ Parameterization:-

- 1) what is Parameterization?
- 2) How to fetch data from Excel sheet?
- 3) Which Jar files are responsible to fetch data from Excel sheet?

⑪ Framework?

- 1) What is DDF?
- 2) What are advantages of DDF?

- 3) what is pom?
- 4) How to design pom class?
- 5) What are getters and setters?
- 6) How to eliminate "state element reference Excep"?
- 7) What is work of initElements method?
- 8) What is Pagefactory?
- 9) Who is responsible to take new address of webpage when we refresh the page?
- 10) When we will get "state element reference Excep"?

13) TestNG:-

- 1) Why TestNG Invented?
- 2) What is TestNG?
- 3) What are advantages of TestNG?
- 4) What is difference b/w JUnit and TestNG?
- 5) What are Annotations in TestNG?
- 6) Explain Annotations workflow?
- 7) How to do parallel execution in TestNG?
- 8) What is Batch execution?
- 9) How to create Emailable report in TestNG?
- 10) What is multibrowsing Testing? How it is possible in TestNG?
- 11) How to run only failed Testcases in TestNG?
- 12) What are the reasons to fail the testcases?
- 13) Can we run Java program without main()?

14) Maven project?

- 1) How to create Maven project?
- 2) What is difference b/w simple project and Maven project?
- 3) What is [Project Object Model] POM.XML?
- 4) What is WebDriver manager?
- 5) Where we add dependencies in Maven project?

b) What are Maven goals?

15) Jenkins :-

- 1) What is Return of Investment?
- 2) What is CI/CD?
- 3) How to schedule Job?

16) Git-Hub

- 1) What is Git and GitHub?
- 2) What is central/Remote Repository?
- 3) How to push and pull the code in GitHub?

17) Cucumber

⑦ What are tags in cucumber?
⑧ What are hooks in cucumber?

- 1) What is cucumber.g? ⑧ What are hooks in cucumber?
- 2) What is Difference b/w TDD and BDD?
- 3) What is meant by feature file?
- 4) What is Runner class?
- 5) What is step definition?
- 6) How to generate Reports in BDD-cucumber?

-
- * What is the work of final, finally, finalize in Java?
 - * What is work of .equals method in string.
 - * What is difference b/w HashMap and Hashtable?

This keyword:- It is used to access global variable from same class.

Ex:-

```
class A {  
    int a = 10;  
    public void test() {  
        int b = 20;  
        System.out.println(b); // 20  
        System.out.println(this.a); // 10  
    }  
}
```

Class B extends A

```
main() {  
    A a = new A();  
    a.test();  
}  
}  $\Rightarrow$  20  
10.
```

Super keyword:- It is used to access global variable from super/different class.

Ex:-

```
class A {  
    int a = 50; // global.  
}  
}
```

Class B extends A

```
int b = 60; // global from var from same class.
```

```
public void test() {  
    System.out.println(this.b); // 60  
    System.out.println(super.a); // 50  
}  
}
```

```
main() {  
    B b = new B();  
    b.test();  
}
```

- ①) ~~final~~ → Keyword for variable no reinitialization
- ②) ~~finally~~ final - for method → No method overriding
- ③) final - class → No inheritance

② finally → Block in exception handling.

③ finalize - method in object class → In which Garbage collector is there.

Static

- 1) They have single copy
- 2) CL is responsible to load static members
- 3) They are load during loading process of class
- 4) Static methods always overload

Nonstatic

- 1) They have multiple copies
- 2) Const* is responsible to load Non-static members
- 3) They will load during object creation
- 4) Non-static methods always overrides.

Upsto x

① ~~Without~~ Without DDF Upsto x ↴

```
main() {
    System.setProperty("webdriver.chrome.driver", "C:\\Users\\Dell\\Downloads\\chromedriver.exe");
    WebDriver driver = new WebDriver();
    // implicit wait
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));
    // delete all cookies
    driver.manage().deleteAllCookies();
    // maximize
    driver.manage().window().maximize();
    // open
    driver.get("https://login-v2.upsto.x.com/");
    // enter UN
    driver.findElement(By.id("username")).sendKeys("2020AL09X");
    // PWD
    driver.findElement(By.id("password")).sendKeys("Pnishi@221");
    // click on login button
    driver.findElement(By.xpath("//div[text()='sign into upsto.x']")).click();
    // DOR
    driver.findElement(By.xpath("//p[@id='yob']")).sendKeys("1995");
    // click on OK
    driver.findElement(By.xpath("//div[text()='No, I'm good']")).click();
    // verify title of page
    String actualTitle = driver.getTitle();
    System.out.println(actualTitle);
    String expectedTitle = "Upsto x PRO";
    if (actualTitle.equals(expectedTitle)) {
        System.out.println("TC passed");
    } else {
        System.out.println("TC failed");
    }
}
```

② with DDF-uptox:-

```
class A {
    main() {
        File file = new File("excelpath");
        WorkbookFactory.create(file).setSheet("sheet1");
        System.setProperty("k", "v");
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
        driver.manage().window().maximize();
        driver.get("uptox");
        // Enter UN
        String UN = driver.findElement(By.xpath("//input[@id='username']")).sendKeys(UN);
        // Enter pWD
        String pWD = driver.findElement(By.xpath("//input[@id='password']")).sendKeys(pWD);
        // Click on Login btn
        driver.findElement(By.xpath("//div[text()='I am not']")).click();
        // Enter pin
        String pin = driver.findElement(By.xpath("//input[@id='pin']")).sendKeys(pin);
        // Click on No I am not
        driver.findElement(By.xpath("//div[text()='No, I am not']")).click();
        // Verify
        String expTitle = driver.getTitle();
        String actTitle = driver.findElement(By.xpath("//span[@id='username']")).getText();
        if (actTitle.equals(expTitle)) {
            System.out.println("TC is pass");
        } else {
            System.out.println("TC is fail");
        }
    }
}
```

~~Pom without DDF :-~~

③ Pom with pagefactory without DDF :-

Class UpstoxLogin1page

// Step - Declaration

@FindBy(xpath = " // i/p[@id = userid")